

## 最小コスト全域木探索を用いたオフライン数式構文認識

江藤 裕子† 福田 亮治† 鈴木 昌和†

† 九州大学大学院数理学研究科 〒 812 福岡市東区箱崎 6-10-1

E-mail:suzuki@math.kyushu-u.ac.jp

‡ 大分大学工学部 〒 870-1192 大分市旦野原 700

E-mail:rfukuda@cc.oita-u.ac.jp

あらまし

現在のOCRシステムは数式を含む科学技術文書に対応できていない。科学技術文書を認識するためには数式構文認識が必要である。本論文では、さまざまな印刷文書に対応でき、1ヶ所の認識誤りが全体に与える影響を少なくするような数式構文認識の手法を提案する。数式構文認識の流れを述べると、まず、各文字や記号に対して、1文字認識の結果と大きさ・位置を元に同一ライン判定を行う。次に、文字や記号を頂点とし、数式の構造を表す接続の候補をコストと向きを持った辺で結び、ネットワークを作成する。最後にネットワークの最小コスト全域木として数式構文の認識結果を取得する。この手法を多くの種類の印刷文書から取り出した数式に適用した認識実験の結果を報告する。

キーワード OCR、数式構文認識、ネットワーク、全域木

## Offline Recognition of the Structure of Mathematical Expressions by Spanning Tree of Minimum Cost

Yuko Eto†

Ryouzi Fukuda†

Masakazu Suzuki†

† Graduate School of Mathematics, Kyushu University 6-10-1 Hakozaki, Higashi-ku, Fukuoka, 812 Japan

E-mail:suzuki@math.kyushu-u.ac.jp

‡ Faculty of Engineering, Oita University 700 Dannoharu, Oita, 870-1192 Japan

E-mail:rfukuda@cc.oita-u.ac.jp

### Abstract

The current systems of OCR usually do not support the recognition of scientific documents including mathematical formulas. For the recognition of scientific documents, it is necessary to recognize the structure of mathematical expressions. In this paper, we propose the method, which is available for various kinds of printings and robust against character recognition errors. The outline of the method is as follows: We first try to classify all characters and symbols into some groups which consist of characters located on the same line. We next construct a network joining characters by possible links of relations with cost. Finally, we obtain a result of the recognition of mathematical formulas as the spanning tree of minimum cost of the network. We report our experimental results for mathematical formulas extracted from various kinds of documents.

key words OCR, recognition of mathematical expressions, network, spanning tree

## 1 はじめに

現在の OCR ソフトは、日本語や英語の文章領域における処理速度や認識精度が大きく向上している。しかし、数式を含む科学技術文書にはまったく対応していない。科学技術文書を認識するには、数学記号の認識と数式部分の構文を理解することが必要になるが、開発には多くの時間とコストがかかるうえに需要が小さいためこのような研究を行っている機関は少ない。

数式構文認識が可能になれば、科学技術文書の電子化において、現在は画像として取り扱われていることが多い数式部分を計算機上で取り扱いやすい形で保存できるようになる。

また、目の不自由な人々の社会進出に大きな役割を果たしている印刷文書の点訳や読み上げは、現在のところ多くのボランティアの人々によっておこなわれている。そこで、OCR を応用して自動点訳や自動読み上げの技術も発展してきているが、数式を含む文書には対応できていない。これに対応するためには、印刷文書における数式の構文認識が必要不可欠である。

一般に、現在の数式認識では多くの場合トップダウン方式が使われているため、一箇所間違った認識をするとそれ以後もすべて崩れてしまう([1][2][3][4])。また、宮崎らの手法では、 $\text{TeX}$  で書かれたものについてはよく認識しているが活字印刷のものには適していない([3])。よって本論文では、部分的に間違っただけの解析をしても後の部分に与える影響が少なく、多様な活字印刷のものに対応できるような手法を提案する。

具体的には、数式中の文字や記号のあいだに親子関係を定めることができると考え、数式全体を向きを持った木として表現する。その実現方法として、文字や記号を頂点とし、可能性のあるすべての親子関係を辺として持つネットワークを構成し、その最小コストの全域木が正しい認識結果になるようにする。たとえば、 $x^{2y}$  という数式に対し、 $x$  と  $2$  は添え字の関係にあり、 $2$  の親は  $x$  である。また、 $2$  と  $y$  は水平に並んでいるので  $y$  の親は  $2$  であるとする。よって、頂点  $x$  から頂点  $2$  への辺と、頂点  $2$  から頂点  $y$  への向きを持った全域木を作ることになる。

本論文では、最初に数式認識を行うための前提条件について述べ、次にその手法を提案する。そして、最後に実験によって得られた評価を述べる。

## 2 前提条件

認識する画像は 400dpi でスキャンした良質画像を用いるものとする。一文字認識エンジンは、能隅

らによって開発されたエンジン ([5]) とあわせて、(株) 東芝の ExpressReader を使わせて頂いた。

### 2.1 一文字認識について

数式認識では一文字認識においてアルファベット、ギリシャ文字、数学記号などが認識できることを想定している。また、認識結果の第二候補までを使って数式認識を考えることにする。一文字認識の結果として認識対象外文字をリジェクトすることがあるが、その場合の第一候補は "リジェクト" とし、第二候補に認識結果の第一候補を入れる。

我々が使用したエンジンでは、連結成分ごとに認識を行い、その後、 $i, j$  などの分離文字や  $=, \leq$  のような連結数が 2 つ以上の数学記号は統合され、一文字として返される。 $Ss, Cc$  のように大文字と小文字が同形のものや、 $Oo0, 11$  については区別せずに返される。大文字と小文字が同形のものについては数式認識をするときにどちらであるかを判定する。 $S, f$  は識別可能であり、 $\sqrt{\quad}$  も認識できる。能隅らのエンジンにおける一文字認識の認識率は、3 冊の書籍からとった一文字画像データベースについて実験した結果、表 1 のようになった。ここでは、アルファベット (Alphabet)、ギリシャ文字 (Greek)、数学記号 (Symbol) の通常サイズ文字 (N) と添え字サイズ文字 (S) について認識率を調べた。アルファベットの文字種はローマン体、イタリック体などの書体を区別している。また、文字数とは実験したデータの個数である。

	文字種	文字数	認識率 (%)
Alphabet(N)	132	5146	96.46
Alphabet(S)	91	1430	96.76
Greek(N)	32	1128	84.73
Greek(S)	19	530	91.57
Symbol(N)	76	2045	83.66
Symbok(S)	20	530	90.96

表 1: 一文字認識の認識率

### 2.2 パラメータの実測

後で述べる同一ライン判定やネットワークの作成において使用する各種のパラメータは、32 冊の本について統計的に調べた数値を用いた。ここでは、その実測値について説明する。

#### 1. 正規化サイズ (NSize)

正規化サイズとは同一ライン上の文字が同じサイズを持つように大きさを補正したものである [2]。ここでは図 1 に示すアセンダー部分

(x) とディセンダー部分 (z) をあわせたサイズとする。実際に測定した正規化サイズの平均と xyz 比の平均を表 2 に示す。

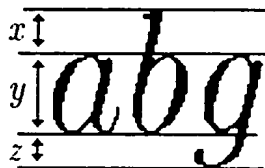


図 1

	NSize	x : y : z
アルファベット	50	278:507:213
ギリシャ文字	48	279:499:191
第一添え字	31	262:545:191
第二添え字	24	249:544:294

表 2: 正規化サイズと xyz 比 (平均値)

## 2. 正規化中心 (NCenter)

正規化中心とは同一ライン上の文字が同じ高さの中心位置を持つように中心位置を補正したものである [2]。ここでは、正規化した矩形の中心を正規化中心とした。

## 3. 文字サイズの比較

テキスト部分のアルファベットの正規化サイズを 1000 としたときの、ギリシャ文字・添え字の正規化サイズ  $H = \frac{h_2}{h_1} \times 1000$  を調べた (図 2)。また、“-” を除く演算子と括弧類の文字サイズ (実画像の高さ) についても同様に調べてみた。その最大値・最小値を表 3、4 に示す。演算子の文字サイズで最小値に近いものは、“=, +” などであった。

## 4. 添え字・演算子・括弧類の中心位置の比較

親文字の正規化サイズを 1000、正規化中心 y 座標を 0 として図 3 に示すパラメータを使って添え字文字の正規化中心位置  $D = \frac{c_1 - c_2}{h} \times 1000$  を調べた (図 3 の  $c_1, c_2$  は、それぞれの文字の正規化中心 y 座標を指す)。また、演算子・括弧類の中心 y 座標についても同様に調べてみた。その最大値・最小値を表 5、6 に示す。

演算子・括弧類を調べた結果、文字サイズは記号によってばらつきがあるが、中心位置は多くの場合アルファベットの正規化中心と近い位置にあることがわかった。

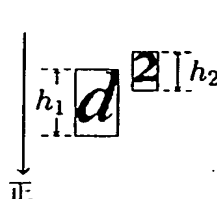


図 2

	max	min
ギリシャ文字	1200	709
第一添え字	750	500
第二添え字	642	387

表 3: アルファベットに対する正規化サイズの最大値・最小値

	max	min
演算子	1100	204
括弧類	1339	888

表 4: アルファベットに対する文字サイズの最大値・最小値

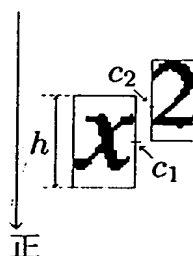


図 3

	max	min
上付き添え字	546	134
下付き添え字	-220	-426

表 5: 添え字の正規化中心位置の最大値・最小値

	max	min
演算子	153	-65
括弧類	108	-93

表 6: 記号の中心位置の最大値・最小値

## 3 処理手順

一文字認識の結果を使って全域木を構成するための手順を述べる。一文字認識では、認識結果の第 1, 2 候補と外接矩形座標 (図 4) を返すものとする。それらを使って、次の順番で処理を行う。

1. 文字に文字種を表すラベルをつける。
2. すべての文字の正規化サイズ・正規化中心を求める。
3. 同一ライン判定を行う。
4. 可能性のあるすべての接続を辺で結んだネットワークを作成する。
5. 最小コスト全域木を構成する。

以下にその詳細を述べる。

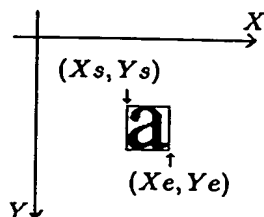


図4:外接矩形座標

### 3.1 文字のラベル付け

認識結果をもとに各文字を次の10種のラベルによって分類する。

- Alp アルファベット、ギリシャ文字、数字
- Ope 二項演算子、関係演算子
- Sum 上下限式の親文字 ( $\sum, \prod, \max, \lim$  など)  
 $\max, \lim$  のように複数文字で構成されているものは同一ライン判定の後で統合する。
- Poi コンマ、ピリオド
- Roo ルート
- Int インテグラル
- Par 括弧
- Fra 分数線  
 認識結果が "—" で、上下の適当な位置に文字があるものを分数線とする。
- Acc アクセント記号  
 認識結果がアクセント候補文字 ( $\hat{x} \hat{x} \overline{PQ}$  のように文字の上につく可能性のある記号) で、その下の適当な位置に文字があるものをアクセント記号とする。
- Oth その他

### 3.2 正規化サイズと正規化中心の設定

外接矩形座標から正規化サイズと正規化中心を2.2で示した数値に基づいて求める。正規化サイズは、文字種ラベル Alp の文字とアセンダーやディセンダーを持つ数学記号については xyz 比 (28:51:21) を使って計算する。演算子は文字そのものの高さの大きいほうをとる。また、コンマ・ピリオドについては前後の文字の正規化サイズのうち大きいほうを採用し、それ以外のは文字そのものの高さとした。正規化中心はすべての文字において、正規化サイズでみたときの文字の中心をとった。

### 3.3 同一ライン判定

同一ライン判定では、数式部分においてさまざまな位置に並んでいる文字や記号を、同じサイズと中心位置を持つ同一ラインクラスに分類し、同一ライン文字の正規化サイズを統一する。

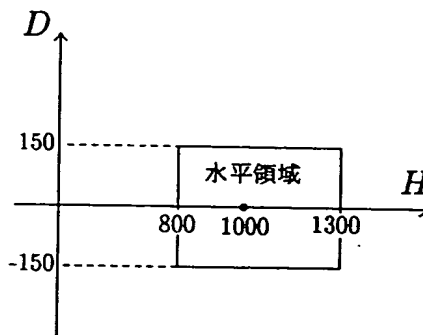


図5:水平領域

#### 3.3.1 クラスタリングの手法

同一ラインクラスのクラスタリングでは、各クラスの基準(後述)の正規化サイズを1000、正規化中心を0として、着目文字の文字サイズ  $H$  と中心位置  $D$  を2.2の3、4で述べた式によって計算する。その計算値が、実測値を元に推定した水平領域(図5)に入るかどうかでクラス分けする。複数のクラスに対して水平領域に入る場合には、 $(H, D)$  が  $(1000, 0)$  に最も近い値をとるクラスにに入れる。どのクラスの水平領域にも入らなかった場合には新しいクラスを作る。このクラスタリングにより、ベースラインの文字・第一添え字のライン・第二添え字のラインなどを取り出すことができる。

#### 3.3.2 クラスタリングの順序と基準の取り方

このクラスタリングでは、基準の正規化サイズを1000として、文字サイズと正規化中心の距離でクラス分けするため、各クラスの基準には正規化サイズが比較的正確に求まる文字を使うことが重要である。よって、各クラスの基準には xyz 比がほぼ一定であるアルファベットと数字を使う。アルファベットや数字が存在しないクラスでは、そのクラスに入るすべての文字の正規化サイズと正規化中心  $y$  座標の平均を基準とする。

基準の取り方から、各文字によって正規化サイズが異なるような文字(分数線や括弧など)はできる限り基準に影響を与えないように、3.1で述べた文字種ラベルの順にクラスタリングを行う。文字種ラベル Alp についてはさらに次のように順序をつける。最初に大文字・小文字の形が異なるアルファベットと "0" 以外の数字をクラスタリングする。次にそのアルファベットと数字を各クラスの

基準として、ギリシャ文字のクラスタリングを行い、最後に“Ss,Cc”のような大文字・小文字が同形のものをクラスタリングする。このとき、大文字と小文字の両方の場合を考えて最適なクラスにいれ、大文字であるか小文字であるかを決定する。そのあとで、残りの文字種ラベルの文字をラベルの順に処理するが、一文字認識の段階でリジェクトされた文字については、第2候補を用いて判定を行うことにする。以上のような順番でクラスタリングを行うことにより、正規化サイズが比較的正確に求まる文字を各クラスの基準として選ぶことができ、より正確な判定ができる。

### 3.3.3 同一ライン判定後の処理

3.1 で示した外接矩形座標から求めた分数線や括弧類などの正規化サイズ・正規化中心は正確とは限らないので、同一ライン判定後に各クラスの基準の正規化サイズ・正規化中心でおきかえる。また、同一ライン判定結果を使って  $\lim, \max$  などを統合し、これ以降一つの文字として扱うことにする。

最後に、各ラインクラスに大きさのレベルを表すサイズラベルをつける。各クラスの基準の正規化サイズによってラベル付けするが、異なるラインクラスでも正規化サイズが近いときには同じラベルになるようにする。

### 3.4 ネットワークの作成

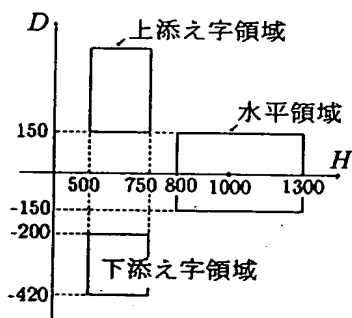


図6: 接続領域

各頂点を接続ラベルとコストを持った有向辺でむすびネットワークを作成する。2つの文字に対して、位置関係と図6に示す水平領域・添え字領域に入るかどうかを調べて接続を考える。この領域は2.2で示した実測値に基づいて定めた。接続ラベルは次の7種類とする。また、文字の大きさの比較は同一ライン判定後につけたサイズラベルによるものとする。

#### 1. 水平接続

- ・同一ラインと判定されたものを左の文字から右の文字へと接続する。

- ・異なるラインで水平領域に入り、あいだに左の文字より大きなサイズラベルの文字が存在しないとき左の文字から右の文字へと接続する。

#### 2. 右上添え字接続

- ・異なるラインで、上添え字領域に入り、あいだに親文字よりも大きなサイズラベルの文字が存在しないとき接続する。

#### 3. 右下添え字接続

- ・異なるラインで、下添え字領域に入り、あいだに親文字よりも大きなサイズラベルの文字が存在しないとき接続する。

#### 4. 上接続

- ・文字種ラベル Sum ( $\Sigma, \cap$  など) の文字に対し、その上にある文字と接続する。

- ・分数線に対し、分子領域にある文字と接続する。

- ・インテグラルに対し、その真上にある文字と接続する。

#### 5. 下接続

- ・文字種ラベル Sum の文字に対し、その下にある文字と接続する。

- ・分数線に対し、分母領域にある文字と接続する。

- ・インテグラルに対し、その真下にくる文字と接続する。

#### 6. ルート接続

- ・ルート記号に対し、その中にある文字と接続する。

#### 7. アクセント接続

- ・アクセント記号に対し、その影響領域の文字と接続する。

コストは外接矩形座標を用いて次のように与える。

- ・水平接続と添え字接続のコストは文字間のスペース  $s$  (図7) とする。

- ・上下限式に対する上接続・下接続やインテグラルの添え字に対するコストは外接矩形の左端座標の差  $d$  (図7) とする。

- ・分数線から分子・分母領域の文字に対する上接続・下接続のコストは外接矩形の左端座標の差と縦方向のスペースの和をとり、分数線の長さを考慮してコストを決める。

- ・ルートとルートの中の文字の接続コストは外接矩形の左端座標の差の  $1/2$  とする。

- ・アクセント接続のコストは縦方向のスペースとする。

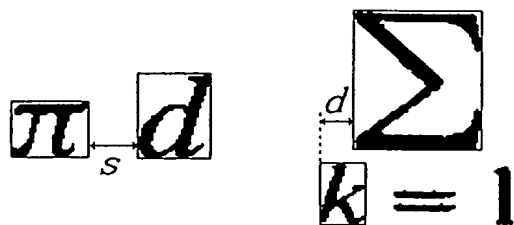


図 7

接続は任意の2組の文字に対して考えるが、親文字の文字種ラベルによって取り得る接続を限定する(表7)。たとえば、“x”を親文字として接続を考えたとき、水平方向と添え字位置のみに文字があると考えられる。

ラベル	取り得る接続
Alp	水平, 添え字接続
Ope	水平接続
Sum	水平, 添え字, 上下接続
Poi	水平接続
Roo	水平, ルート接続
Int	水平, 添え字, 上下接続
Par	水平, 添え字 (閉じ括弧のみ) 接続
Fra	水平, 上下接続
Acc	水平, 添え字, アクセント接続
Oth	水平, 添え字接続

表 7: 文字種ラベルと接続

最後に、全体をみて以下の調整を行い、ネットワークを完成させる。

- ・ルートの影響領域の文字と影響領域外の文字の接続はルート記号を介すように調整する。
- ・アクセント記号の影響領域の文字と影響領域外の文字の接続はアクセント記号を介すように調整する。
- ・分子分母領域から分数の外への接続を分数線からの接続に変える。
- ・文字種ラベル Sum の上下接続の影響領域を調整する。

### 3.5 最小コスト全域木の構成

可能性のあるすべての頂点を接続したネットワークから各頂点の親を決めて、数式の構造を表す全域木を構成する。数式として矛盾のない木とは、各

頂点が1つの接続方向(表7)に対して2つ以上の子をもたない木である。

3.4で生成したネットワークのすべての全域木を考えて、数式としての条件を満たす全域木の中でコストが最小のものを選びたいが、計算量が多くなるため次のような方法で、一つの全域木を得ることにする。

1. 親候補を一つしか持っていない頂点について、その親子関係を採用する。
2. 採用した辺と同じ始点で同じラベルの辺があれば、それらを取り除く。
3. 親が決まっていなかった頂点について、最小コストの親と接続する。

数式の構造と接続の条件からネットワークにはループが存在しないので、ネットワークが連結ならば、この方法で1つの最小コスト全域木ができる。このとき、全域木と同じ始点で同じラベルの辺が存在しなければ、矛盾のないひとつの数式構造が認識できたことになる。しかし、そのような辺が存在するときには数式として矛盾した結果となる。ネットワークが連結でない場合は、連結正分数と同じ数の木に分かれて結果が返される。

## 4 実験結果と考察

16冊の本からとった103行の数式について数式認識実験を行ったところ、72行については数式として正しい結果を返したが、残りの31行は数式として矛盾のある結果となった。

誤りの主な原因は次のようなものであった。

- ・一文字認識において、接触文字などで文字を正しく切り出せなかった。(5/31)
- ・一文字認識における誤認識により、正規化サイズや正規化中心が変わり、頂点間の接続領域が実際と異なる領域になった。(9/31)
- ・一文字認識における誤認識により、文字種ラベルが実際の画像と異なり、数式としての木を構成できなかった。(7/31)
- ・同一ライン判定のときに、“Ss,Cc,Pp”などの大文字、小文字の判定を誤った。(6/31)
- ・コンマと  $p'$  などの“ $'$ ”を区別できなかった。(3/31)
- ・分数線の取り出しや、アクセント記号の判定を誤った。(3/31)

次にいくつかの認識結果を示す(例1,2,3,4,5)。

例1は小文字の“j”を大文字の“J”と誤認識しているが数式としての構文は正しく認識している。例2では、“x”を“χ”と誤認識している

が、各頂点からの水平接続を一つに制限していることにより正しく認識できている。これらは誤認識により正規化サイズをあやまってもそれを補って数式構文を正しく認識している例である。例3では、“G”を“a”と誤認識したことにより、添え字接続が水平接続になった。例4では“p”の大文字、小文字判定を間違ったことと、同一ライン判定の結果、ルートとルートの中の“2”(添え字)が同一ラインとなり、ルートの正規化サイズが正しく求められなかったために誤認識している。例5では、“オーバーライン”とその下の文字列との間のスペースが広いと、アクセント記号の判定を誤って“マイナス”としている。

## 5 今後の課題

認識実験の結果から、今後次のような改良を加えていく。

- ・アセンダー、ディセンダーを持つ文字の正規化サイズを求めるときに、現在は文字サイズによらず一定のxyz比を用いているが、添え字サイズになると割合が変わってくることが多いので文字サイズによってxyz比を検討する。
- ・同一ライン判定において“Ss、Cc、Pp”のように大文字と小文字が同形の文字の判定を行っているが、ラインだけでは決定できないことが多くあるので、大文字と小文字の両方の場合を考慮してネットワークを作り、数式を表す木として最適なほうを選ぶようにする。
- ・一文字認識において誤認識は避けられないので、第二候補も加えたネットワークを考えて最適な全域木が得られるようにする。
- ・記号類の正規化サイズの求め方や、同一ライン判定に改良の余地がある。
- ・左添え字、条件式、行列などへの対応を加える。

## 6 まとめ

本論文では、数式の可能性をネットワークで表し、その最小コスト全域木を求めることにより数式を認識するという手法を提案した。さまざまな文書から得た文字サイズなどの平均値をパラメータとして使った結果、多くの種類の印刷文書に対応できることがわかった。また、トップダウン方式で問題になる部分的な誤認識が後の認識結果に影響を与えるということに対しては、誤認識をカバーして正しい数式を認識できることもあるが、不十分な結果も多い。

今後、同一ライン判定結果などを使って、数式認識結果を検証することや、数式の基準線上の文

字を取り出すことなどを組み込んで、多くの数式に正確に対応できるように改善することが必要である。

## 参考文献

- [1] 岡本 正行、トワキョンドムサフィリ ハシム: 「周辺分布特徴を用いた数式構造認識」、信学論、J78-D-II、No.2、pp366-370(1995-2)
- [2] 岡本 正行、東 裕之: 「記号レイアウトに注目した数式構造認識」、信学論、J-78D-II、No.3、pp474-482(1995-3)
- [3] 鈴木昌和、玉利文和、井上浩一、宮崎亮乃輔、宮平彩乃: 「OCRを用いた科学技術文書の自動点訳について」、信学技法、HCS97-8、pp7-14(1997-9)
- [4] K.Inoue、R.Miyazaki、M.Suzuki: 「Optical recognition of printed mathematical documents」、Proceedings of the Third Asian Technology Conference in Mathematics、pp280-289、(1998-8)
- [5] 能隅 進一、福田 亮治、玉利 文和、鈴木 昌和: 「絞り込み法による数式文字認識とその日本語/数式領域切り出しへの応用」、電子情報通信学会論文誌、J-83D-II、No.3、pp1-12(2000-3) 掲載予定

$$f^* = f_{N+1} - \sum_{j=1}^N c_j x^{\deg(f_{N+1}) - \deg(f_j)} f_j$$

$$f^* = f_{N+1} - \sum_{j=1}^N c_j x^{\deg(f_{N+1}) - \deg(f_j)} f_j$$

例 1. 入力画像 (上) と認識結果 (下)

$$\Phi_{\eta}(x) = \sum \langle \pi_{-\infty}(x) \xi_j, \eta \rangle e_j = \sum \langle \xi_j, \pi_{\infty}(x)^{-1} \eta \rangle e_j.$$

$$\Phi_{\eta}(\chi) = \sum \langle \pi_{-\infty}(x) \xi_j, \eta \rangle e_j = \sum \langle \xi_j, \pi_{\infty}(x)^{-1} \eta \rangle e_j.$$

例 2. 入力画像 (上) と認識結果 (下)

$$\langle x, x \rangle_c = \gamma_1 \bar{\xi}_1 \xi_1 + \cdots + \gamma_s \bar{\xi}_s \xi_s$$

$$\langle x, x \rangle_a = \gamma_1 \bar{\xi}_1 \xi_1 + \cdots + \gamma_l \bar{\xi}_l \xi_l$$

例 3. 入力画像 (上) と認識結果 (下)

$$p - \sqrt{1+p^2} = \frac{-1}{p + \sqrt{1+p^2}} = -\frac{1}{C_1} e^{-a_1}$$

$$P^{-\sqrt{1+p^2}} = \frac{-1}{p + \sqrt{1+p^2}} = -\frac{1}{C_1} e^{-a_1}$$

例 4. 入力画像 (上) と認識結果 (下)

$$R(\alpha(\varepsilon); x, w) = \frac{(\varphi(x) - \varphi(w)) \overline{\varphi(w)}}{\lambda - \alpha(\varepsilon)} + F(\alpha(\varepsilon); x, w) - F(\alpha(\varepsilon); w, w),$$

$$R(\alpha(\varepsilon); x, w) = \frac{(\varphi(x) - \varphi(w))^{-1} \overline{\varphi(w)}}{\lambda - \alpha(\varepsilon)} + l(\alpha(\varepsilon); x, w) - l(\alpha(\varepsilon); w, w),$$

例 5. 入力画像 (上) と認識結果 (下)